Improved impossible differential cryptanalysis on 7-round ARIA

Xiaoning Feng[1] and Hongyu Wu[1*]

[1*]College of Computer Science and Technology, Harbin Engineering University, Nantong Street, Harbin, 150001, Heilongjiang, China.

*Corresponding author(s). E-mail(s): B221060008@hrbeu.edu.cn;
Contributing authors: fengxiaoning@hrbeu.edu.cn;

Abstract

The ARIA algorithm is an important Square structure block cipher algorithm, which has become the block cipher standard in South Korea. Our paper proposes a new method for automatically exploring impossible differential paths across four rounds, leading to the discovery of previously unexplored and unrecognized impossible differential paths. Moreover, our investigation yielded a novel conditional distinguisher specific to the ARIA algorithm. Synthesizing the undiscovered 4-round impossible differential path with conditional distinguisher, we introduces a impossible differential attack method targeting for the 7-round ARIA algorithm. Within the scope of this study, we conducted a comprehensive analysis of the time complexity associated with the proposed novel impossible differential attack method under both classical and quantum contexts, employing the early abort technique. Our contribution also encompasses the proposal of a 7-round quantum impossible differential attack method targeting ARIA algorithms. Remarkably, our research findings attest to the distinct advantages conferred by the impossible differential attack method within the quantum environment setting, surpassing the average attack complexity of prior attack algorithms. Conclusions drawn from our paper remind cryptographic designers to be mindful of cryptographic security in quantum environment setting.

Keywords: ARIA algorithm, Impossible differential attack, Early abort technique, Grover algorithm, Automated search, Quantum nested search

1 Introduction

The ARIA algorithm [1] was published by National Security Research Institute of Korea. The algorithm adopts the same Square structure as AES and the design principle is similar to AES. Designers analyze a variety of popular attacks, such as linear attack, differential attack and Square attack. The designer claims that the ARIA algorithm has higher security than AES. In 2004, the ARIA algorithm was selected as the block cipher standard in South Korea.

The two most widely used and important cryptanalysis in block cipher are differential attacks and linear attacks. The impossible differential attack [2] is a crucial technique within the differential attack family and has been extensively applied in cryptanalysis of Square structures such as AES[3, 4]. The impossible differential attack differs from the conventional notion of differential attacks in that it relies on the fact that the probability of obtaining differential paths is zero when the correct key is used to encrypt the plaintext. The impossible differential attack leverages the differential paths with probability zero (typically constructed through the miss-in-the-middle technique) and eliminates all the keys that produce the differential paths with probability zero. After filtering out a large number of possible key values, the correct key value is obtained.

Related Work. Impossible differential paths play a pivotal role in the execution of impossible differential attacks, as they serve as distinctive routes that lead to diverse attack strategies[5–7]. However, the process of manually identifying these paths proves to be exceedingly challenging due to its reliance on heuristic reasoning and intricate derivation. While the Mixed integer linear programming (MILP) [8] method can be employed for the search process of differential paths, there are also some disadvantages associated with MILP. The MILP method involves mathematical modeling and solving complex linear programming problems. When multiple rounds of cryptanalysis are involved, the MILP problem may become too large, and it is easy to cause too many inequalities, resulting in high computing costs. The MILP method may need to simplify the problem or the cryptographic system, which may lead to inaccurate results or less differential paths to find, making it difficult to carry out further analysis.

Acquiring impossible differential attacks is confronted with the issue of excessive computational complexity. Notably, in recent years, a range of optimization techniques has emerged as viable strategies for improving the efficiency of impossible differential attacks. These techniques encompass the early abort technique[9], calculation tables[10], and so on. The essence of the early abort technique lies in its multi-layer nested search structure, which systematically identifies and filters out plaintext pairs that have already reached an expired state at each layer of the nested search. The progressive advancements in recent years have established the early abort technique as a prominent frontrunner in the optimization of impossible differential attacks. Our statistical analysis reveals that a substantial majority of research endeavors concerning impossible differential attacks prioritize the incorporation of early abort technique to enhance their efficiency and effectiveness.

Recently, numerous researchers have worked on the security of symmetric cryptography in the quantum environment[11–17]. The aforementioned research suggests that quantum computing can achieve a significant speedup in the field of symmetric

cryptanalysis, leading to the potential for successful attacks against many symmetric cryptosystems. Within the context of our discussion, a cryptanalytic process that employs quantum algorithms is termed a quantum environment setting, while a process that does not use quantum algorithms is referred to as a classical environment setting.

There are several quantum implementations of classical differential techniques, including the quantization of the Boomerang attack [18], the quantization of the counting-based differential attack [19], and the quantization of table-based differential attack [20]. Intuitively, the nested search process of the early abort technique can be accelerated by a quadratic factor through quantum computing model.

Our Contributions. Previous research has encountered two primary challenges. Firstly, the acquisition of impossible differential paths has proven difficult. Secondly, the impact of the quantum computing model on classical attack methods has been overlooked. Consequently, the main objective of this study is to develop a novel attack method targeting the ARIA algorithm, aiming to overcome these challenges and achieve groundbreaking research outcomes. Specifically, our research contributes significantly in the following aspects:

1. Development of an automated program and discovery of a new 4-round impossible differential path. We have developed an automated program based on the miss-in-the-middle technique, which has successfully facilitated the search and identification of a previously undiscovered 4-round impossible differential path for the ARIA algorithm. This path has not been detected in previous research on impossible differential attacks on the ARIA algorithm, and its validity has been rigorously verified.
2. Discovery of ARIA's condition distinguisher and novel attack method. We have made a significant breakthrough by identifying a condition distinguisher in the ARIA algorithm, which exhibits a probability of occurrence that surpasses the random case by a factor of 2^{48}. By combining the condition distinguisher with the newly discovered 4-round impossible differential path, we have developed a novel impossible differential attack method specifically designed to target the 7-round ARIA algorithm.
3. Complexity analysis utilizing early abort technique. Our study incorporates the early abort technique to conduct a comprehensive analysis of the time complexity for the proposed novel impossible differential attack method in both classical and quantum environments. We also present a 7-round quantum impossible difference method specifically designed for the ARIA algorithm. Our research demonstrates that in the quantum environment, the impossible differential attack method exhibits significant superiority compared to previous attack algorithms.

Organization. Section 2 provides an overview of the necessary background knowledge, including the ARIA algorithm, impossible differential attack, and relevant symbol representation. We demonstrate the automatic impossible differential path discovery procedure and the new impossible differential paths discovered in Section 3. In Section 4, we demonstrate conditional distinguisher and novel impossible differential attacks. In Section 5, we conducted complexity analysis of our attack in both classical

and quantum environments, and we also compare the complexity of our attack with other known attacks on the ARIA algorithm. Section 6 concludes this paper.

2 Preliminaries

2.1 ARIA algorithm

The ARIA algorithm is an SPN-type block cipher with a block length of 128-bit. The ARIA algorithm provides support for three distinct key lengths: 128-bit, 192-bit, and 256-bit. The number of iteration rounds corresponding to the variable key length is 12-round, 14-round and 16-round. The encryption process of a complete ARIA algorithm is shown in Figure 1. The ARIA algorithm consists primarily of three components: Round Key Addition (RKA), Substitution Layer (SL) and Diffusion Layer (DL). It should be noted that in the final round of the ARIA algorithm, the DL operation is replaced by RKA. The 128-bit plaintext, ciphertext, and intermediate state of the ARIA algorithm are generally regarded as a 4×4 byte matrix, and the indices of the matrix is shown in Figure 2.

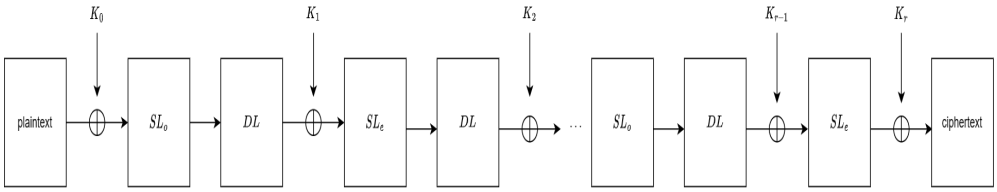

Fig. 1 ARIA algorithm diagram

The specific components of the ARIA algorithm are described as follows:

1. RKA: XOR the input state of each round with the round key k_i. The round key k_i is generated from the master key using the key expansion algorithm. This paper focuses on the process of recovering the round keys without considering the key expansion algorithm.
2. SL: The ARIA algorithm employs S_1, S_1^{-1}, as well as S_2, S_2^{-1}, to form two alternating S-boxes, namely SL_o and SL_e. SL_o and SL_e are utilized alternately in the odd and even rounds, respectively. The implementations of SL_o and SL_e are depicted in Figure 2.

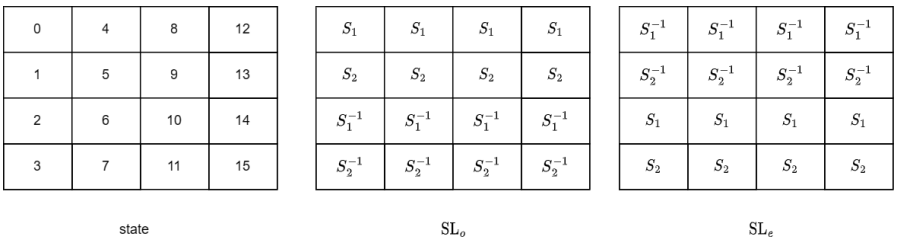

Fig. 2 SL layer illustration of the ARIA algorithm

3. DL: the diffusion layer of the ARIA algorithm maps a 16-byte input to a 16-byte output. In particular, the diffusion layer is an involution operation with $\mathsf{DL} = \mathsf{DL}^{-1}$. A linear function can be represented as follows:

$$
\begin{aligned}
y_0 &= x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{14} & y_8 &= x_0 \oplus x_1 \oplus x_4 \oplus x_7 \oplus x_{10} \oplus x_{13} \oplus x_{15} \\
y_1 &= x_2 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{12} \oplus x_{15} & y_9 &= x_0 \oplus x_1 \oplus x_5 \oplus x_6 \oplus x_{11} \oplus x_{12} \oplus x_{14} \\
y_2 &= x_1 \oplus x_4 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{15} & y_{10} &= x_2 \oplus x_3 \oplus x_5 \oplus x_6 \oplus x_8 \oplus x_{13} \oplus x_{15} \\
y_3 &= x_0 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{13} \oplus x_{14} & y_{11} &= x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14} \\
y_4 &= x_0 \oplus x_2 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{14} \oplus x_{15} & y_{12} &= x_1 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{11} \oplus x_{12} \\
y_5 &= x_1 \oplus x_3 \oplus x_4 \oplus x_9 \oplus x_{10} \oplus x_{14} \oplus x_{15} & y_{13} &= x_0 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{13} \\
y_6 &= x_0 \oplus x_2 \oplus x_7 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{13} & y_{14} &= x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_9 \oplus x_{11} \oplus x_{14} \\
y_7 &= x_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{13} & y_{15} &= x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{15}
\end{aligned}
$$

2.2 Impossible differential attack

Impossible differential attack usually divides the encryption block $E : \{0,1\}^n \to \{0,1\}^n$ of r round into three parts, $E = E_{out} \circ E_{imp} \circ E_{in}$, as shown in Figure 3. The rounds of $E_{out} \circ E_{imp} \circ E_{in}$ are respectively r_{out}, r_{imp}, r_{in}, there is $r = r_{out} + r_{imp} + r_{in}$. Impossible differential attack first needs to construct an impossible differential path $\Delta_X \nrightarrow_{r_{imp}} \Delta_Y$ in the middle round E_{imp}, that is, the probability of propagating from input difference Δ_X to output difference Δ_Y in r_{imp} rounds is 0. The previous and subsequent rounds r_{out}, r_{in} are used for key recovery of encrypted blocks. Define D_{in} as the plaintext input difference set and D_{out} as the ciphertext output difference set. Plaintext pair $p = (x, y)$ satisfy $x \oplus y \in D_{in} \wedge E_k(x) \oplus E_k(y) \in D_{out}$. If a plaintext input difference in D_{in} propagates to Δ_X through E_{in} and a ciphertext output difference in D_{out} propagates to Δ_Y through E_{out}, we can discard candidate keys u satisfying the following equation:

$$(E_{in}(u)(x) \oplus E_{in}(u)(y) = \Delta x) \wedge (E_{out}^{-1}(u)(E_k(x)) \oplus E_{out}^{-1}(u)(E_k(y)) = \Delta y)$$

We use K_{in} and K_{out} to denote the subkeys guessed at rounds E_{in} and E_{out}, respectively. The goal of impossible differential attack is to discard as many key values as possible, so that the possible value range $K_{in \cup out}$ of the retained key is as small as possible. Impossible differential attack generally includes the following two stages:

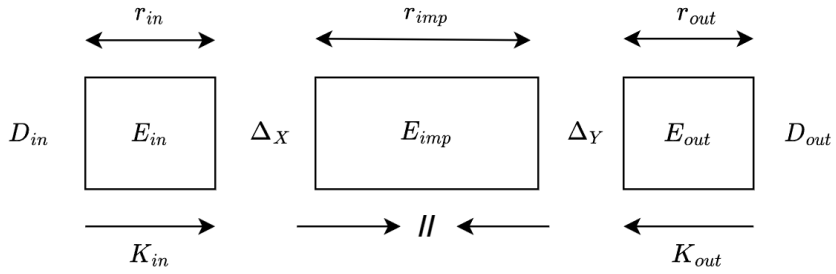

Fig. 3 Impossible differential attack

1. Plaintext pair collection phase. Find the plaintext pair $p = (x, y)$, the plaintext pair satisfies the formula $x \oplus y \in D_{in} \wedge E_k(x) \oplus E_k(y) \in D_{out}$.
2. Plaintext pair filtering stage. k_i and plaintext pairs form filter conditions. The early abort technique is usually used in this process to filter out the plaintext pairs that do not meet the conditions by traversing the subkey k_i in $K_{in \cup out}$.

Finally, $K_{in \cup out}$ is removed since the differential Δ_X to Δ_Y is impossible.

Below we provide a brief description of some commonly used notations in this paper. In this paper, we use several notations to describe the ARIA algorithm and the impossible differential attack. Specifically, P represents plaintext, C represents ciphertext, $(P, P^{'})$ and $(C, C^{'})$ represent plaintext pairs and ciphertext pairs, m represents the intermediate state, Δm represents the difference of m. Additionally, $P^{(i)}$ and $C^{(i)}$ represent the i-th byte of plaintext and ciphertext respectively. The notation of $k_{i,(p,\ldots,r)}$ represents the bytes from p to r of the i-th round subkey. In particular, the DL process is replaced by an RKA process in the last round of the ARIA algorithm, resulting in two RKA processes in the final round. The notation of $m^{\mathsf{RKA/SL/DL}}_{i,(p,\ldots,r)}$ refers to the intermediate value obtained after the RKA/SL/DL process in the i-th round , where the bytes at position p to r of the intermediate value.

3 A Method for Automatically Searching Impossible Differential Paths

This study introduces a novel automatic search technique based on the miss-in-the-middle approach [21] to identify 4-round impossible differential paths in the ARIA cipher. Specifically, this method leverages the propagation of forward and backward differences in the intermediate state of the ARIA cipher, where certain pairs of cipher blocks become equal or unequal after each propagation, respectively. This contradiction ultimately leads to the discovery of an impossible differential path. The proposed algorithm is outlined in detail in Algorithm 1.

In the above algorithm, x refers to the number of cipher blocks with non-zero differential value among the cipher blocks resulting from the first round of the ARIA algorithm, while y refers to the number of cipher blocks at the end of the fourth round whose differential value is non-zero and equal. Instantiating x and y into a set (as and

Algorithm 1 Obtain impossible differential paths

Input: x, y, 4-round ARIA algorithm
Output: Impossible differential paths

$as \Leftarrow x$ ▷ Instantiate x as a collection as
$ds \Leftarrow$ 2-round ARIA(as) ▷ After 2-round of ARIA algorithm, from as to get ds
$S, S', res \Leftarrow \Phi$ ▷ Initialize the collections of S,S',res
for each element di of ds **do**
    for each cipher block C_i of di (a total of 16) **do**
        if two block cipher blocks (i_1, i_2) are equal between C_i **then**
            $S+ = (i_1, i_2)$
        end if
    end for
end for
$fs \Leftarrow y$ ▷ Instantiate y as a collection fs
$dxs \Leftarrow$ 2-round ARIA$^{-1}(fs)$ ▷ After 2-round of reverse ARIA algorithm, from fs to get dxs
for each element di of dxs **do**
    for each cipher block C_i of di **do**
        if two block cipher blocks (i_1, i_2) are not equal between C_i **then**
            $S'+ = (i_1, i_2)$
        end if
    end for
end for
for each element ai of as **do**
    for each element fi of fs **do**
        if the S corresponding to ai is equal to the S' corresponding to fi **then**
            $res+ = (ai, fi)$
        end if
    end for
end for
return res ▷ Return the Impossible differential paths

fs) means generating all possible values of x and y based on the given constraints and collecting them into a set. The collection ds is obtained by applying 2-round of ARIA algorithm on each element in the set as. Similarly, The collection dxs is obtained by applying 2-round of reverse ARIA algorithm on each element in the set fs. Through separate iterations of the internals of ds and dxs, store the identical blocks from the second round's internals in S and the distinct blocks from the third round's initial internals in $S^{'}$. We store the final result in the res collection. This approach allows us to enumerate all possible pairs of initial and final differential paths that satisfy the constraints, which is necessary for further analysis and search. We have made the above-mentioned automatic algorithm available on Github [22] for interested readers to explore further.

We employed the aforementioned algorithm to automatically discover multiple instances of impossible differential paths, including all 4-round impossible differential

paths found in the existing literature. For example, the impossible differential path proposed in [5]:

$$(0,0,0,0,a_4,0,0,0,0,a_9,0,0,0,0,0,0) \nrightarrow_{4-round} (0,0,0,0,f,0,f,0,0,0,0,0,0,f,f,0)$$

The impossible differential path of [6]:

$$(a_0,0,0,0,0,a_5,0,0,0,0,0,0,0,0,0,0) \nrightarrow_{4-round} (f,f,0,0,0,0,f,0,0,0,0,0,0,0,f,0)$$

The impossible differential path of [7]:

$$(0,0,0,0,0,0,0,a_7,0,0,0,0,0,0,a_{13},0,0) \nrightarrow_{4-round} (f,0,0,0,f,0,0,0,0,0,0,0,0,f,0,f)$$

Naturally, we also discovered numerous new impossible differential paths that had not been analyzed before. It is worth noting that there is still a lack of further research on the relationship between the number of impossible differential paths and variables such as x and y. However, this is not the main focus of our research in this paper. In this paper, we use a new impossible differential path (obtained by $x = 2, y = 8$) for our impossible differential attack. See Theorem 1 for details.

Theorem 1 (Impossible differential path)**.** *As shown in the Figure 4, There exist plaintext pairs that satisfy the input difference condition: the difference at byte positions* $(1, 11)$ *is non-zero, while the differences at all other byte positions are zero. After undergoing 4 rounds of the ARIA transformation, the resulting ciphertext pairs satisfy the output difference condition: the differences at byte positions* $(0, 1, 2, 4, 6, 9, 10, 14)$ *are non-zero and equal, while the differences at all other byte positions are zero. This impossible differential path is expressed by:*

$$(0,a_1,0,0,0,0,0,0,0,0,0,a_{11},0,0,0,0) \nrightarrow_{4-round} (f,f,f,0,f,0,f,0,0,f,f,0,0,0,f,0)$$

Among them, a_1*,* a_{11}*,* f *are any non-zero values.*

Proof. First, we start the derivation from the first 2-round forward. The RKA operation does not change the difference, and the SL modifies the difference corresponding to the position of the cipher block. The input state of the plaintext pair satisfies $(0,a_1,0,0,0,0,0,0,0,0,0,a_{11},0,0,0,0)$, where a_1 and a_{11} are non-zero values. After RKA and SL operation, the difference becomes $(0,b_1,0,0,0,0,0,0,0,0,0,b_{11},0,0,0,0)$, where b_1 and b_{11} are non-zero bytes. After the first round of DL operation, the difference becomes $(0,0,b_1 \oplus b_{11}, b_{11}, b_{11}, b_1, 0, b_1 \oplus b_{11}, b_1, b_1 \oplus b_{11}, 0, 0, b_1 \oplus b_{11}, 0, b_{11}, b_1)$. After the second round of RKA operation and SL operation, the difference becomes $(0,0,c_2,c_3,c_4,c_5,0,c_7,c_8,c_9,0,0,c_{12},0,c_{14},c_{15})$. After the second round of DL operation, the difference becomes $(d_0,d_1,d_2,d_3,d_4,d_5,d_6,d_7,d_8,d_9,d_{10},d_{11},d_{12},d_{13},d_{14},d_{15})$. According to the operation of DL, there is $d_6 = d_{12} = c_2 \oplus c_7 \oplus c_9 \oplus c_{12}$.

Subsequently, we initiate the deduction from the reverse process of 4-round ARIA algorithm using the initial state $(f,f,f,0,f,0,f,0,0,f,f,0,0,0,f,0)$. After a DL^{-1} transformation, the difference becomes $(0,0,0,f,f,f,0,0,0,0,0,0,0,f,0,0)$.

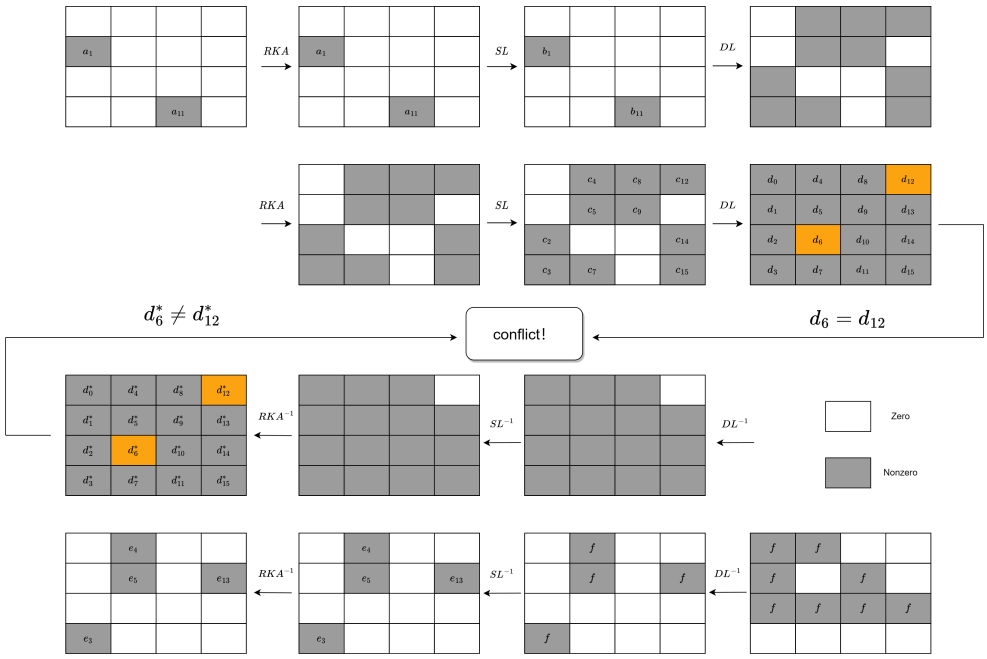

Fig. 4 The 4-round impossible differential path

After another SL^{-1} and RKA^{-1} transformation, the difference is transformed into $(0,0,0,e_3,e_4,e_5,0,0,0,0,0,0,0,e_{13},0,0)$, where e_3,e_4,e_5,e_{13} are non-zero bytes. Then after another DL^{-1} operation, the difference becomes $(e_3 \oplus e_4 \oplus e_{13}, e_5, e_4, e_5 \oplus e_{13}, e_5, e_3 \oplus e_4, e_{13}, e_3 \oplus e_{13}, e_4 \oplus e_{13}, e_5, e_3 \oplus e_5 \oplus e_{13}, e_3 \oplus e_4, 0, e_3 \oplus e_{13}, e_3 \oplus e_4 \oplus e_5, e_4 \oplus e_5)$. Finally, after SL^{-1} and RKA^{-1} operations, the difference is $(d_0^*, d_1^*, d_2^*, d_3^*, d_4^*, d_5^*, d_6^*, d_7^*, d_8^*, d_9^*, d_{10}^*, d_{11}^*, d_{12}^*, d_{13}^*, d_{14}^*, d_{15}^*)$. $d_6^* \neq 0$ because there is $\mathsf{SL}^{-1}(d_6^*) = e_{13} \neq 0$. However, there is $d_{12}^* = 0$. Therefore, the contradiction $d_6^* \neq d_{12}^*$ arises, so the above 4-round impossible differential path is established. □

4 Impossible differential attack on 7-round ARIA

Below we introduce a conditional distinguisher that we discovered on the ARIA algorithm. Compared with the random situation, the conditional distinguisher can show an advantage in 2^{48} probability (the probability of the random case is 2^{-80}, and the probability of the conditional distinguisher is 2^{-32}). For detailed information, please refer to Theorem 2 for details.

Theorem 2 (Conditional distinguisher)**.** *As shown in the R1 of Figure 5, when transforming from m_1^{SL} to m_1^{DL} through the first round of diffusion layer, if m_1^{SL} satisfies the following four equations, then the probability that the difference of m_1^{DL} at $(0,1,2,6,7,9,10,11,12,13)$ is zero is 2^{-32}. However, this probability is 2^{-80} in the*

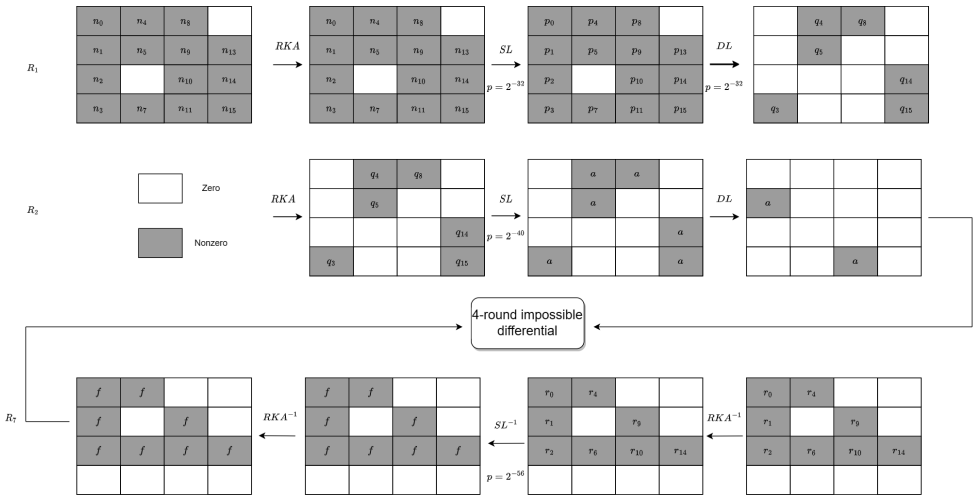

Fig. 5 The 7-round impossible differential attack

random case. The four equations are:

$$\begin{aligned} p_3 &= p_9 \\ p_7 &= p_{13} \\ p_2 &= p_8 \\ p_0 \oplus p_4 \oplus p_5 \oplus p_{10} \oplus p_{14} \oplus p_{15} &= 0 \end{aligned} \tag{1}$$

Proof. First, we define the ciphertext space for m_1^{DL} such that the difference of m_1^{DL} at $(0, 1, 2, 6, 7, 9, 10, 11, 12, 13)$ is zero and the difference in other bytes is non-zero (the size of the space is 2^{48}). Assuming the existence of such a ciphertext space at m_1^{DL}, applying DL^{-1} to this space yields the following relations: $p_3 = p_9 = q_5 \oplus q_{14}$, $p_7 = p_{13} = q_3 \oplus q_8$ and $p_2 = p_8 = q_4 \oplus q_{15}$. The key point is that the Formula 1 holds with probability 1, regardless of the values of $q_3, q_4, q_5, q_8, q_{14}, q_{15}$. And due to the following formula:

$$\begin{cases} p_0 = q_3 \oplus q_4 \oplus q_8 \oplus q_{14} \\ p_4 = q_5 \oplus q_8 \oplus q_{14} \oplus q_{15} \\ p_5 = q_3 \oplus q_4 \oplus q_{14} \oplus q_{15} \\ p_{10} = q_3 \oplus q_5 \oplus q_8 \oplus q_{15} \\ p_{14} = q_3 \oplus q_4 \oplus q_5 \oplus q_{14} \\ p_{15} = q_4 \oplus q_5 \oplus q_8 \oplus q_{15} \end{cases} \tag{2}$$

There is $p_0 \oplus p_4 \oplus p_5 \oplus p_{10} \oplus p_{14} \oplus p_{15} = 0$. The four equations are established with probability 2^{-8} randomly. Hence, the number of m_1^{SL} at $2^{112} \times (-2^{-8})^4 = 2^{80}$. Since DL is a linear transformation and the space size at m_1^{DL} is 2^{48}, the probability of transforming from m_1^{SL} to m_1^{DL} is $2^{48}/2^{80} = 2^{-32}$. □

According to the two theorems we proved above, we can extend the 4-round impossible differential path to a 7-round impossible differential path by adding two rounds in front of the 4-round impossible differential path and one round at the end. So the main attack process can be summarized as follows based on the obtained 7-round impossible differential path:

1. Choose structures of 2^{112} plaintexts that they are different at the 14 types $(1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15)$ taking $0 \sim 2^8$ in the above 14 bytes. Thus, the plaintext space has $2^{112} \times 2^{112} \times 1/2 = 2^{223}$ plaintext pairs.
2. Take 2^N of the above structures(2^{N+112} plaintexts and 2^{N+223} plaintext pairs), we only retain the pairs that the corresponding ciphertext pairs are zero difference at the 8 bytes $(3, 5, 7, 8, 11, 12, 13, 15)$. Then, there are $2^{223+N} \times 2^{-8\times(16-8)} = 2^{N+159}$ plaintext pairs (ciphertext pairs).
3. Guess the partial key $(k_{8,0}, k_{8,1}, k_{8,2}, k_{8,4}, k_{8,6}, k_{8,9}, k_{8,10}, k_{8,14})$ of the key k_8. We calculate the following formula for each retained ciphertext pairs $(C, C^{'})$:

$$
\begin{aligned}
\Delta m_{7,0}^{\mathsf{SL}^{-1}} &= S_1^{-1}(C^{(0)} \oplus k_{8,0}) \oplus S_1^{-1}(C^{'(0)} \oplus k_{8,0}) \\
\Delta m_{7,1}^{\mathsf{SL}^{-1}} &= S_2^{-1}(C^{(1)} \oplus k_{8,1}) \oplus S_2^{-1}(C^{'(1)} \oplus k_{8,1}) \\
\Delta m_{7,2}^{\mathsf{SL}^{-1}} &= S_1(C^{(2)} \oplus k_{8,2}) \oplus S_1(C^{'(2)} \oplus k_{8,2}) \\
\Delta m_{7,4}^{\mathsf{SL}^{-1}} &= S_1^{-1}(C^{(4)} \oplus k_{8,4}) \oplus S_1^{-1}(C^{'(4)} \oplus k_{8,4}) \\
\Delta m_{7,6}^{\mathsf{SL}^{-1}} &= S_1(C^{(6)} \oplus k_{8,6}) \oplus S_1(C^{'(6)} \oplus k_{8,6}) \\
\Delta m_{7,9}^{\mathsf{SL}^{-1}} &= S_2^{-1}(C^{(9)} \oplus k_{8,9}) \oplus S_2^{-1}(C^{'(9)} \oplus k_{8,9}) \\
\Delta m_{7,10}^{\mathsf{SL}^{-1}} &= S_1(C^{(10)} \oplus k_{8,10}) \oplus S_1(C^{'(10)} \oplus k_{8,10}) \\
\Delta m_{7,14}^{\mathsf{SL}^{-1}} &= S_1(C^{(14)} \oplus k_{8,14}) \oplus S_1(C^{'(14)} \oplus k_{8,14})
\end{aligned}
$$

   Choose ciphertext pairs satisfying $\Delta m_{7,0}^{\mathsf{SL}^{-1}} = \Delta m_{7,1}^{\mathsf{SL}^{-1}} = \Delta m_{7,2}^{\mathsf{SL}^{-1}} = \Delta m_{7,4}^{\mathsf{SL}^{-1}} = \Delta m_{7,6}^{\mathsf{SL}^{-1}} = \Delta m_{7,9}^{\mathsf{SL}^{-1}} = \Delta m_{7,10}^{\mathsf{SL}^{-1}} = \Delta m_{7,14}^{\mathsf{SL}^{-1}}$. At this point, the number of remaining ciphertext pairs is $2^{N+159} \times 2^{-56} = 2^{N+103}$.
4. Note that the plaintext pair corresponding to the ciphertext pair retained in the above steps is $(P, P^{'})$. Guess the 14 bytes of k_1, and filter out the intermediate state that satisfies the four equations in Theorem 2, so that the remaining intermediate state pairs can then be analyzed in the following steps.
   (a) Guess the value of the key $(k_{1,3}, k_{1,9})$ and calculate:

$$
\begin{aligned}
\Delta m_{1,3}^{\mathsf{SL}} &= S_2^{-1}(P^{(1)} \oplus k_{1,3}) \oplus S_2^{-1}(P^{'(1)} \oplus k_{1,3}) \\
\Delta m_{1,9}^{\mathsf{SL}} &= S_2(P^{(9)} \oplus k_{1,9}) \oplus S_2(P^{'(9)} \oplus k_{1,9})
\end{aligned}
$$

   Choose plaintext pairs that satisfies $\Delta m_{1,3}^{\mathsf{SL}} = \Delta m_{1,9}^{\mathsf{SL}}$, then there are $2^{N+103} \times 2^{-8} = 2^{N+95}$ plaintext pairs remaining.

(b) Guess the value of the key $(k_{1,7}, k_{1,13})$ and calculate:

$$\Delta m_{1,7}^{\mathsf{SL}} = S_2^{-1}(P^{(7)} \oplus k_{1,7}) \oplus S_2^{-1}(P'^{(7)} \oplus k_{1,7})$$
$$\Delta m_{1,13}^{\mathsf{SL}} = S_2(P^{(13)} \oplus k_{1,13}) \oplus S_2(P'^{(13)} \oplus k_{1,13})$$

Choose plaintext pairs that satisfies $\Delta m_{1,7}^{\mathsf{SL}} = \Delta m_{1,13}^{\mathsf{SL}}$, then there are $2^{N+95} \times 2^{-8} = 2^{N+87}$ plaintext pairs remaining.

(c) Guess the value of the key $(k_{1,2}, k_{1,8})$ and calculate:

$$\Delta m_{1,2}^{\mathsf{SL}} = S_1^{-1}(P^{(2)} \oplus k_{1,2}) \oplus S_1^{-1}(P'^{(2)} \oplus k_{1,2})$$
$$\Delta m_{1,8}^{\mathsf{SL}} = S_1(P^{(8)} \oplus k_{1,8}) \oplus S_1(P'^{(8)} \oplus k_{1,8})$$

Choose plaintext pairs that satisfies $\Delta m_{1,2}^{\mathsf{SL}} = \Delta m_{1,8}^{\mathsf{SL}}$, then there are $2^{N+87} \times 2^{-8} = 2^{N+79}$ plaintext pairs remaining.

(d) Guess the value of the key $(k_{1,0}, k_{1,4}, k_{1,5}, k_{1,10}, k_{1,14}, k_{1,15})$ and calculate:

$$\Delta m_{1,0}^{\mathsf{SL}} = S_1(P^{(0)} \oplus k_{1,0}) \oplus S_1(P'^{(0)} \oplus k_{1,0})$$
$$\Delta m_{1,4}^{\mathsf{SL}} = S_1(P^{(4)} \oplus k_{1,4}) \oplus S_1(P'^{(4)} \oplus k_{1,4})$$
$$\Delta m_{1,5}^{\mathsf{SL}} = S_2(P^{(5)} \oplus k_{1,5}) \oplus S_2(P'^{(5)} \oplus k_{1,5})$$
$$\Delta m_{1,10}^{\mathsf{SL}} = S_1^{-1}(P^{(10)} \oplus k_{1,10}) \oplus S_1^{-1}(P'^{(10)} \oplus k_{1,10})$$
$$\Delta m_{1,14}^{\mathsf{SL}} = S_1^{-1}(P^{(14)} \oplus k_{1,14}) \oplus S_1^{-1}(P'^{(14)} \oplus k_{1,14})$$
$$\Delta m_{1,15}^{\mathsf{SL}} = S_2^{-1}(P^{(15)} \oplus k_{1,15}) \oplus S_2^{-1}(P'^{(15)} \oplus k_{1,15})$$

Choose plaintext pairs that satisfies $\Delta m_{1,0}^{\mathsf{SL}} \oplus \Delta m_{1,4}^{\mathsf{SL}} \oplus \Delta m_{1,5}^{\mathsf{SL}} \oplus \Delta m_{1,10}^{\mathsf{SL}} \oplus \Delta m_{1,14}^{\mathsf{SL}} \oplus \Delta m_{1,15}^{\mathsf{SL}} = 0$, then there are $2^{N+79} \times 2^{-8} = 2^{N+71}$ plaintext pairs remaining.

(e) Guess the value of the key $(k_{1,1}, k_{1,11})$ and calculate:

$$\Delta m_{1,1}^{\mathsf{SL}} = S_2(P^{(1)} \oplus k_{1,1}) \oplus S_1(P'^{(1)} \oplus k_{1,1})$$
$$\Delta m_{1,11}^{\mathsf{SL}} = S_2^{-1}(P^{(11)} \oplus k_{1,11}) \oplus S_2^{-1}(P'^{(11)} \oplus k_{1,11})$$

(f) Calculate $\Delta m_1^{\mathsf{DL}} = \mathsf{DL}(\Delta m_1^{\mathsf{SL}})$. Choose plaintext pairs whose $(0, 1, 2, 6, 7, 9, 10, 11, 12, 13)$ bytes of Δm_1^{DL} are 0. According to the conclusion of the proved Theorem 2, the probability of this is 2^{-32}. So there are $2^{N+71} \times 2^{-32} = 2^{N+39}$ remaining plaintext pairs.

5. Guess the value of the key $(k_{2,3}, k_{2,4}, k_{2,5}, k_{2,8}, k_{2,14}, k_{2,15})$ and calculate:

$$\Delta m_{2,3}^{\mathsf{SL}} = S_2(m_{2,3} \oplus k_{2,3}) \oplus S_2(m_{2,3} \oplus k_{2,3})$$
$$\Delta m_{2,4}^{\mathsf{SL}} = S_1^{-1}(m_{2,4} \oplus k_{2,4}) \oplus S_1^{-1}(m_{2,4} \oplus k_{2,4})$$
$$\Delta m_{2,5}^{\mathsf{SL}} = S_2^{-1}(m_{2,5} \oplus k_{2,5}) \oplus S_2^{-1}(m_{2,5} \oplus k_{2,5})$$
$$\Delta m_{2,8}^{\mathsf{SL}} = S_1^{-1}(m_{2,8} \oplus k_{2,8}) \oplus S_1^{-1}(m_{2,8} \oplus k_{2,8})$$
$$\Delta m_{2,14}^{\mathsf{SL}} = S_1(m_{2,14} \oplus k_{2,14}) \oplus S_1(m_{2,14} \oplus k_{2,14})$$
$$\Delta m_{2,15}^{\mathsf{SL}} = S_2(m_{2,15} \oplus k_{2,15}) \oplus S_2(m_{2,15} \oplus k_{2,15})$$

Choose plaintext pairs that satisfies $\Delta m_{2,3}^{\mathsf{SL}} = \Delta m_{2,4}^{\mathsf{SL}} = \Delta m_{2,5}^{\mathsf{SL}} = \Delta m_{2,8}^{\mathsf{SL}} = \Delta m_{2,14}^{\mathsf{SL}} = \Delta m_{2,15}^{\mathsf{SL}}$. The above attack is an impossible differential attack. The guessed subkey bytes are all wrong, which is the key value that needs to be excluded.

5 Complexity analysis

5.1 Classic environment setting

Our attack has a data complexity of 2^{N+112} (number of chosen plaintexts). We record the time complexity of the calculation in the following Table 1. Note that the computational complexity of the encryption operation is equivalent to that of the decryption operation. We consider ARIA's three encryption components, namely RKA, SL, and DL, as one-third rounds. We use common early abort techniques to reduce complexity. The main idea of the early abort technique is to partition the keys that need to be guessed into a set of states and to guess only some of the states, thereby reducing the number of plaintext pairs that are already invalid in advance. For instance, in the process of guessing the key k_8 in the step 3, we initially verify whether $\Delta m_{7,0}^{\mathsf{SL}^{-1}}$ and $\Delta m_{7,1}^{\mathsf{SL}^{-1}}$ are same. After eliminating some plaintext pairs, we then examine whether $\Delta m_{7,1}^{\mathsf{SL}^{-1}}$ and $\Delta m_{7,2}^{\mathsf{SL}^{-1}}$ are same. The detailed complexity analysis is as follows:

- In step 3, the k_8 value of 8 bytes needs to be guessed, and there are two S-box involved in the operation in the SL^{-1} layer. Carry out two operations of RKA^{-1} and SL^{-1}, and performing a complete formula operation is equivalent to performing 2/3 times of 1 round of operations. Therefore, $(2^{N+159} \times 2^{16} + 2^{N+151} \times 2^{24} + 2^{N+143} \times 2^{32} + 2^{N+135} \times 2^{40} + 2^{N+127} \times 2^{48} + 2^{N+119} \times 2^{56} + 2^{N+111} \times 2^{64}) \times 2 \times 8/16 \times 2/3 = 14/3 \times 2^{N+175}$ 1-round operations are required here.
- In step 4.(a), the k_1 value of 2 bytes needs to be guessed, and there are two S-box involved in the operation in the SL layer. Therefore, $2^{64} \times 2^{N+103} \times 2^{16} \times 2 \times 2/16 \times 2/3 = 2/3 \times 2^{N+181}$ 1-round operations are required here.
- In step 4.(b), the k_1 value of 2 bytes needs to be guessed, and there are two S-box involved in the operation in the SL layer. Therefore, $2^{64} \times 2^{16} \times 2^{N+95} \times 2^{16} \times 2 \times 2/16 \times 2/3 = 2/3 \times 2^{N+189}$ 1-round operations are required here.
- In step 4.(c), the k_1 value of 2 bytes needs to be guessed, and there are two S-box involved in the operation in the SL layer. Therefore, $2^{64} \times 2^{16} \times 2^{16} \times 2^{N+87} \times 2^{16} \times 2 \times 2/16 \times 2/3 = 2/3 \times 2^{N+197}$ 1-round operations are required here.

- In step 4.(d), the k_1 value of 6 bytes needs to be guessed, and there are six S-box involved in the operation in the SL layer. Therefore, $2^{64} \times 2^{16} \times 2^{16} \times 2^{16} \times 2^{N+79} \times 2^{48} \times 2 \times 6/16 \times 2/3 = 2^{N+238}$ 1-round operations are required here.
- In step 4.(e), the k_1 value of 2 bytes needs to be guessed, and there are two S-box involved in the operation in the SL layer. Therefore, $2^{64} \times 2^{16} \times 2^{16} \times 2^{16} \times 2^{48} \times 2^{N+71} \times 2^{16} \times 2 \times 2/16 \times 2/3 = 2/3 \times 2^{N+245}$ 1-round operations are required here.
- In step 4.(f), there is no need to guess the key value, only the DL operation is performed. Therefore, $2^{64} \times 2^{16} \times 2^{16} \times 2^{16} \times 2^{48} \times 2^{16} \times 2^{N+71} \times 2 \times 1/3 = 2/3 \times 2^{N+247}$ 1-round operations are required here.
- In step 5, the k_2 value of 6 bytes needs to be guessed, and there are six S-box involved in the operation in the SL layer. Therefore, $(2^{64} \times 2^{16} \times 2^{16} \times 2^{16} \times 2^{48} \times 2^{16}) \times (2^{N+39} \times 2^{16} + 2^{N+31} \times 2^{24} + 2^{N+23} \times 2^{32} + 2^{N+15} \times 2^{40} + 2^{N+7} \times 2^{48}) \times 2 \times 6/16 \times 2/3 = 5 \times 2^{N+230}$ 1-round operations are required here.

Table 1 The complexity of each step in the 7-round impossible differential attack

Step	Guessed round key	Guess the number of bits in the key	Matching conditions	Number of remaining pairs	Time complexity
3	$(k_{8,0}, k_{8,1}, k_{8,2}, k_{8,4}$ $k_{8,6}, k_{8,9}, k_{8,10}, k_{8,14})$	64	56	$2^{N+159} \times 2^{-56} = 2^{N+103}$	$14/3 \times 2^{N+175}$
4.(a)	$(k_{1,3}, k_{1,9})$	16	8	$2^{N+103} \times 2^{-8} = 2^{N+95}$	$2/3 \times 2^{N+181}$
4.(b)	$(k_{1,7}, k_{1,13})$	16	8	$2^{N+95} \times 2^{-8} = 2^{N+87}$	$2/3 \times 2^{N+189}$
4.(c)	$(k_{1,2}, k_{1,8})$	16	8	$2^{N+87} \times 2^{-8} = 2^{N+79}$	$2/3 \times 2^{N+197}$
4.(d)	$(k_{1,0}, k_{1,4}, k_{1,5}$ $k_{1,10}, k_{1,14}, k_{1,15})$	48	8	$2^{N+79} \times 2^{-8} = 2^{N+71}$	2^{N+238}
4.(e)	$(k_{1,1}, k_{1,11})$	16	0	$2^{N+71} \times 2^{0} = 2^{N+71}$	$2/3 \times 2^{N+245}$
4.(f)	-	-	32	$2^{N+71} \times 2^{-32} = 2^{N+39}$	$2/3 \times 2^{N+247}$
5	$(k_{2,3}, k_{2,4}, k_{2,5}$ $k_{2,8}, k_{2,14}, k_{2,15})$	48	40	$2^{N+39} \times 2^{-40} = 2^{N-1}$	$5 \times 2^{N+230}$

So the total number of matching conditions is $(56+8+8+8+8+32+40) = 160$-bit. The number of bits of our candidate key is $(64+16+16+16+48+16+48) = 224$-bit. By analyzing 2^{N+159} plaintext pairs, the probability of the key value remaining is:

$$(1 - 2^{-160})^{2^{N+159}} \backsimeq e^{-(2^{N+159}) \times 2^{-160}}$$

So there are still $(2^{224} - 1) \times e^{-(2^{N-1})}$ wrong values for keys remaining. Let $(2^{224} - 1) \times e^{-(2^{N-1})} < 1$, then there is $N = 9$.

According to the above analysis, we conclude that the total time complexity required by our attack is: $(14/3 \times 2^{9+175} + 2/3 \times 2^{9+181} + 2/3 \times 2^{9+189} + 2/3 \times 2^{9+197} + 2^{9+238} + 2/3 \times 2^{9+245} + 2/3 \times 2^{9+247} + 5 \times 2^{9+230}) \times 1/7 \approx 2^{253}$.

5.2 Quantum environment setting

Using the Grover algorithm can significantly accelerate the search process in cryptanalysis(For a comprehensive introduction to Grover's principle, readers are encouraged to consult [23]). Martin et al.[24] applied the Grover algorithm to search for keys with different weights in a side-channel attack, achieving a quadratic speedup. Biasse et al. [25] incorporated classical search and preprocessing techniques into the Grover search algorithm to enhance its efficiency for cryptanalysis purposes. The acceleration of the cryptanalysis process can be further enhanced by combining search problems. Schrottenloher conducted a separate evaluation of the two cases of nested search in his PhD thesis. One is the nested search of test conditions, while the other involves nested search of the search space (early abort technique being a type of nested search space)[26]. Schrottenloher drew a brief conclusion that when the search space is nested, the complexity of the search space will be reduced from the classic $|K_1|(T_1+|K_2|(T_2+|K_3|(T_3+...+|K_l|T_l)))$ to $\sqrt{|K_1|}(T_1+\sqrt{|K_2|}(T_2+\sqrt{|K_3|}(T_3+...+\sqrt{|K_l|T_l})))$, where K_i denotes the key space to be searched and T_i denotes the table space to be searched at each level.

Building upon Schrottenloher's quantum nesting algorithm, we extend its application to incorporate early abort technique. Subsequently, we proceed to present the quantized implementation of our impossible differential attack on the ARIA algorithm, integrating early abort technique. The proposed algorithm is outlined as Algorithm 2.

Algorithm 2 Quantum Impossible Differential Attack for ARIA Algorithm

Input: 2^{N+159} plaintext pairs

Output: finds the only key $k = (k_{8,0}, k_{8,1}, k_{8,2}, k_{8,4}, k_{8,6}, k_{8,9}, k_{8,10}, k_{8,14}, k_{1,3}, k_{1,9}, k_{1,7}$
$k_{1,13}, k_{1,2}, k_{1,8}, k_{1,0}, k_{1,4}, k_{1,5}, k_{1,10}, k_{1,14}, k_{1,15}, k_{1,1}, k_{1,11},$
$k_{2,3}, k_{2,4}, k_{2,5}, k_{2,8}, k_{2,14}, k_{2,15})$

Initialize the all output key state as the $|0\rangle$ state.
$H^{\otimes 16}$ acts on $|0\rangle^{\otimes 16}$ to prepare the equal weight superposition state $|k_{8,0}, k_{8,1}\rangle$.
Repeat the Grover iteration S times($S \approx \frac{\pi}{4}\sqrt{16}$) **do**
    Compute $\Delta m_{7,0}^{\mathsf{SL}^{-1}}, \Delta m_{7,1}^{\mathsf{SL}^{-1}}$.
    Obtain plaintext pairs that satisfy $\Delta m_{7,0}^{\mathsf{SL}^{-1}} = \Delta m_{7,1}^{\mathsf{SL}^{-1}}$.
    $H^{\otimes 8}$ acts on $|0\rangle^{\otimes 8}$ to prepare the equal weight superposition state $|k_{8,2}\rangle$.
    Repeat the Grover iteration S times($S \approx \frac{\pi}{4}\sqrt{8}$) **do**
        Compute $\Delta m_{7,1}^{\mathsf{SL}^{-1}}, \Delta m_{7,2}^{\mathsf{SL}^{-1}}$.
        Obtain plaintext pairs that satisfy $\Delta m_{7,1}^{\mathsf{SL}^{-1}} = \Delta m_{7,2}^{\mathsf{SL}^{-1}}$.
        ...
    end Repeat
end Repeat
Return k ▷ Return the only key

Our algorithm ignores the Grover diffusion operator (for simplicity of the algorithm) and the measurement process (due to the principle of deferred measurements[27], the measurement process can be moved to the end). The inner

search can be regarded as the unitary operator of the outer search, The unitary operator is divided into two parts. The first part is to obtain the plaintext pairs that meet the conditions. The second part is the search process for the remaining keys. In the last layer of search, when the built table is empty, start flipping $k_{2,15}$. Specifically, we utilize Corollary 2-"Quantum early abort technique complexity theorem" from [28] to support our analysis of the complexity of the quantum impossible differential attack on the ARIA algorithm.

Theorem 3. ***Quantum early abort technique complexity theorem.*** *There exists a quantum algorithm $\mathcal{G}$, quantum algorithm $\mathcal{G}$ quantizes the early abort technique of impossible differential attack, and discovers a correct key $(k_1, k_2, ..., k_l)$ using the following time complexity:*

$$2N\sum_{j=1}^{l}(t_j+t)(\prod_{m=1}^{j-1}\sigma_m)(\frac{\pi}{2})^j\sqrt{\prod_{m=1}^{j-1}|K_m|}$$

Where N represents the size of the initial plaintext; t_i represents the time to evaluate the condition T_i; t represents the operation time to perform copy or exchange on the register (quantum case); σ_m indicates the probability of filtering the plaintext for the m-th time; $|K_m|$ represents the key space size of $k \in K_m$.

We use the conclusions to further analyze the time complexity of our impossible differential attack. When taking into account that the complexity of the classical implementation circuit is equivalent to that of the quantum implementation circuit, we can further analyze the complexity as follows:

- In step 3, $2 \times (2^{168}(\frac{\pi}{2})\sqrt{2^{16}} + 2^{160}(\frac{\pi}{2})^2\sqrt{2^{24}} + 2^{152}(\frac{\pi}{2})^3\sqrt{2^{32}} + 2^{144}(\frac{\pi}{2})^4\sqrt{2^{40}} + 2^{136}(\frac{\pi}{2})^5\sqrt{2^{48}} + 2^{128}(\frac{\pi}{2})^6\sqrt{2^{56}} + 2^{120}(\frac{\pi}{2})^7\sqrt{2^{64}}) \times 2 \times 8/16 \times 2/3 \approx 2^{177}$ 1-round operations are required here.
- In step 4.(a), $2 \times (2^{112}(\frac{\pi}{2})^8\sqrt{2^{64+16}}) \times 2 \times 2/16 \times 2/3 \approx 2^{156}$ 1-round operations are required here.
- In step 4.(b), $2 \times (2^{104}(\frac{\pi}{2})^9\sqrt{2^{64+16+16}}) \times 2 \times 2/16 \times 2/3 \approx 2^{156}$ 1-round operations are required here.
- In step 4.(c), $2 \times (2^{96}(\frac{\pi}{2})^{10}\sqrt{2^{64+16+16+16}}) \times 2 \times 2/16 \times 2/3 \approx 2^{157}$ 1-round operations are required here.
- In step 4.(d), $2 \times (2^{88}(\frac{\pi}{2})^{11}\sqrt{2^{64+16+16+16+48}}) \times 2 \times 6/16 \times 2/3 \approx 2^{175}$ 1-round operations are required here.
- In step 4.(e), $2 \times (2^{80}(\frac{\pi}{2})^{12}\sqrt{2^{64+16+16+16+48+16}}) \times 2 \times 2/16 \times 2/3 \approx 2^{175}$ 1-round operations are required here.
- In step 4.(f), there is no need to guess the key value that belongs to the same layer of search as 4.(e). $2 \times (2^{80}(\frac{\pi}{2})^{12}\sqrt{2^{64+16+16+16+48+16}}) \times 2 \times 1/3 \approx 2^{176}$ 1-round operations are required here.
- In step 5, $2 \times (2^{48}(\frac{\pi}{2})^{13}\sqrt{2^{176+16}} + 2^{40}(\frac{\pi}{2})^{14}\sqrt{2^{176+16+16}} + 2^{32}(\frac{\pi}{2})^{15}\sqrt{2^{176+16+16+16}} + 2^{24}(\frac{\pi}{2})^{16}\sqrt{2^{176+16+16+16+16}} + 2^{16}(\frac{\pi}{2})^{17}\sqrt{2^{176+16+16+16+16+16}}) \times 2 \times 6/16 \times 2/3 \approx 2^{156}$ 1-round operations are required here.

To summarize, under the quantum environment setting, the complexity of our attack is $(2^{177} + 2 \times 2^{156} + 2^{157} + 2 \times 2^{175} + 2^{176} + 2^{156}) \times 1/7 \approx 2^{175}$.

Finally, we compare our attack scheme with other attack schemes. In particular, we compute the average complexity of the attack time:

$$\text{Average attack complexity} = \frac{\text{Total time complexity}}{\text{Attack bits}}$$

Table 2 lists the comparison between the attack scheme in this paper and previous attack.

Table 2 Comparison of Impossible Differential Attacks on ARIA

Paper	Attack round	Attack bits	Total time complexity	Average attack complexity	Data complexity
[29]	6	96	2^{112}	2^{16}	2^{121}
[5]	7	192	2^{219}	2^{27}	2^{120}
[6]	7	192	2^{210}	2^{18}	2^{118}
[7]	7	160	2^{190}	2^{30}	2^{117}
[30]	7	224	2^{248}	2^{24}	2^{119}
[31]	7	192	2^{218}	2^{26}	2^{119}
Classic environment	7	224	2^{253}	2^{29}	2^{121}
Quantum environment	7	224	2^{175}	2^{-49}	2^{121}

6 Conclusion

This research paper presents a novel and successful impossible differential attack on the 7-round ARIA algorithm, leveraging a newly discovered 4-round impossible differential path and conditional distinguisher. The utilization of automated programs for impossible differential paths and the identification of new impossible differential paths contribute to a deeper understanding of the vulnerabilities inherent in ARIA algorithms. Employing the proposed quantum impossible difference method for the ARIA algorithm, our 7-round impossible difference attack exhibits a time complexity of 2^{175} and a data complexity of 2^{121}. Notably, our attack scheme surpasses existing methods in terms of average attack complexity, establishing itself as the most efficient approach. This methodology can be considered as a quantum variant of our proposed classical impossible difference method. Although our quantum impossible differential attack cannot produce the effect of quadratical acceleration, it has still made substantial progress ($2^{175} \ll 2^{253}$). Our conclusions illustrate that the security of ciphers in the classical environment does not guarantee their immunity in the quantum environment setting.

Acknowledgments. The author would like to thank my PhD supervisor prof. dr. Feng for providing me with relatively free research space and useful feedback. He would also like to express his appreciation towards the reviewers and the associate editor for comments that helped improve the manuscript.

Declarations

- **Funding**: This work is supported by the National Natural Science Foundation of China (No.51979048).
- **Conflict of interest**: The authors have no conflicts of interest to declare that are relevant to the content of this article.
- **Availability of data and materials**: Any datasets used and/or analysed during the current study that have not been included in this published article or its supplementary information files are available from the corresponding author on reasonable request.

References

[1] Kwon, D., Kim, J., Park, S., Sung, S.H., Sohn, Y., Song, J.H., Yeom, Y., Yoon, E.-J., Lee, S., Lee, J., *et al.*: New block cipher: Aria. In: Information Security and Cryptology-ICISC 2003: 6th International Conference, Seoul, Korea, November 27-28, 2003. Revised Papers 6, pp. 432–445 (2004). Springer

[2] Mollimard, V.: Algorithms for differential cryptanalysis. PhD thesis, Universite Rennes (2022)

[3] Jithendra, K., Shahana, T.: New results in related key impossible differential cryptanalysis on reduced round aes-192. In: 2018 International Conference On Advances in Communication and Computing Technology (ICACCT), pp. 1–5 (2018). IEEE

[4] Jiang, Z., Jin, C., Wang, Z.: Multiple impossible differentials attack on aes-192. IEEE Access **7**, 138011–138017 (2019)

[5] Su, C.: New impossible deferential attack on 7-round reduced aria. Journal of Computer Applications **32**(01), 45 (2012)

[6] Shen, X., He, J.: Improved impossible differential attack on 7-round reduced aria-256. KSII Transactions on Internet and Information Systems (TIIS) **13**(11), 5773–5784 (2019)

[7] Jiang, Z., Jin, C.: Multiple impossible differentials cryptanalysis on 7-round aria-192. Security and Communication Networks **2018** (2018)

[8] Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Information Security and Cryptology: 7th International Conference, Inscrypt 2011, Beijing, China, November 30–December 3, 2011. Revised Selected Papers 7, pp. 57–76 (2012). Springer

[9] Zhang, K., Lai, X., Guan, J., Hu, B.: Research on the security level of μ^2 against impossible differential cryptanalysis. KSII Transactions on Internet and

Information Systems (TIIS) **16**(3), 972–985 (2022)

[10] Tolba, M., Abdelkhalek, A., Youssef, A.M.: Impossible differential cryptanalysis of reduced-round skinny. In: Progress in Cryptology-AFRICACRYPT 2017: 9th International Conference on Cryptology in Africa, Dakar, Senegal, May 24-26, 2017, Proceedings, pp. 117–134 (2017). Springer

[11] Malviya, A.K., Tiwari, N., Chawla, M.: Quantum cryptanalytic attacks of symmetric ciphers: A review. Computers and Electrical Engineering **101**, 108122 (2022)

[12] Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round feistel cipher and the random permutation. In: 2010 IEEE International Symposium on Information Theory, pp. 2682–2685 (2010). IEEE

[13] SANTOLI, T., SCHAFFNER, C.: Using simon's algorithm to attack symmetric-key cryptographic primitives. Quantum Information and Computation **17**(1&2), 0065–0078 (2017)

[14] Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II 36, pp. 207–237 (2016). Springer

[15] Leander, G., May, A.: Grover meets simon–quantumly attacking the fx-construction. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II 23, pp. 161–178 (2017). Springer

[16] Naya-Plasencia, M., Schrottenloher, A.: Optimal merging in quantum k-xor and k-sum algorithms. In: Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II 30, pp. 311–340 (2020). Springer

[17] Schrottenloher, A.: Improved quantum algorithms for the k-xor problem. In: Selected Areas in Cryptography: 28th International Conference, Virtual Event, September 29–October 1, 2021, Revised Selected Papers, pp. 311–331 (2022). Springer

[18] Frixons, P., Naya-Plasencia, M., Schrottenloher, A.: Quantum boomerang attacks and some applications. In: International Conference on Selected Areas in Cryptography, pp. 332–352 (2022). Springer

[19] Denisenko, D.: Quantum differential cryptanalysis. Journal of Computer Virology

and Hacking Techniques, 1–8 (2022)

[20] Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Quantum differential and linear cryptanalysis. IACR Transactions on Symmetric Cryptology **2016**(1), 71–94 (2016)

[21] Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In: Advances in Cryptology—EUROCRYPT'99: International Conference on the Theory and Application of Cryptographic Techniques Prague, Czech Republic, May 2–6, 1999 Proceedings 18, pp. 12–23 (1999). Springer

[22] Hongyu, W.: Automatically find impossible differential paths. https://github.com/whyaza/Automatically-find-impossible-differential-paths (2022)

[23] Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)

[24] Martin, D.P., Montanaro, A., Oswald, E., Shepherd, D.: Quantum key search with side channel advice. In: Selected Areas in Cryptography–SAC 2017: 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers 24, pp. 407–422 (2018). Springer

[25] Biasse, J.-F., Pring, B.: A framework for reducing the overhead of the quantum oracle for use with grover's algorithm with applications to cryptanalysis of sike. Journal of Mathematical Cryptology **15**(1), 143–156 (2020)

[26] Schrottenloher, A.: Quantum algorithms for cryptanalysis and quantum-safe symmetric cryptography. PhD thesis, Sorbonne université (2021)

[27] Gurevich, Y., Blass, A.: Quantum circuits with classical channels and the principle of deferred measurements. Theoretical Computer Science **920**, 21–32 (2022)

[28] David, N., Naya-Plasencia, M., Schrottenloher, A.: Quantum impossible differential attacks: applications to aes and skinny. Designs, Codes and Cryptography, 1–29 (2023)

[29] Wu, W., Zhang, W., Feng, D.: Impossible differential cryptanalysis of reduced-round aria and camellia. Journal of computer science and technology **22**(3), 449–456 (2007)

[30] Ou, H., Wang, X., Li, Y., Lei, Y.: A new impossible difference path and corresponding attack for aria algorithm. Journal of Cryptologic Research **7**(4), 8 (2020)

[31] Xie, G., Wei, H.: Impossible differential attack of block cipher aria. Journal of